Automatic Selection of Context Configurations for Improved Class-Specific Word Representations

Ivan Vulić[1], **Roy Schwartz**[2,3], **Ari Rappoport**[4]
Roi Reichart[5], **Anna Korhonen**[1]
[1] Language Technology Lab, DTAL, University of Cambridge
[2] CS & Engineering, University of Washington and [3]Allen Institute for AI
[4] Institute of Computer Science, The Hebrew University of Jerusalem
[5] Faculty of Industrial Engineering and Management, Technion, IIT
{iv250,alk23}@cam.ac.uk roysch@cs.washington.edu
arir@cs.huji.ac.il roiri@ie.technion.ac.il

Abstract

This paper is concerned with identifying contexts useful for training word representation models for different word classes such as adjectives (A), verbs (V), and nouns (N). We introduce a simple yet effective framework for an automatic selection of *class-specific context configurations*. We construct a context configuration space based on universal dependency relations between words, and efficiently search this space with an adapted beam search algorithm. In word similarity tasks for each word class, we show that our framework is both effective and efficient. Particularly, it improves the Spearman's ρ correlation with human scores on SimLex-999 over the best previously proposed class-specific contexts by 6 (A), 6 (V) and 5 (N) ρ points. With our selected context configurations, we train on only 14% (A), 26.2% (V), and 33.6% (N) of all dependency-based contexts, resulting in a reduced training time. Our results generalise: we show that the configurations our algorithm learns for one English training setup outperform previously proposed context types in another training setup for English. Moreover, basing the configuration space on universal dependencies, it is possible to *transfer* the learned configurations to German and Italian. We also demonstrate improved per-class results over other context types in these two languages.

1 Introduction

Dense real-valued word representations (embeddings) have become ubiquitous in NLP, serving as invaluable features in a broad range of tasks (Turian et al., 2010; Collobert et al., 2011; Chen and Manning, 2014). The omnipresent `word2vec` skip-gram model with negative sampling (SGNS) (Mikolov et al., 2013) is still considered a robust and effective choice for a word representation model, due to its simplicity, fast training, as well as its solid performance across semantic tasks (Baroni et al., 2014; Levy et al., 2015). The original SGNS implementation learns word representations from local bag-of-words contexts (BOW). However, the underlying model is equally applicable with other context types (Levy and Goldberg, 2014a).

Recent work suggests that "not all contexts are created equal". For example, reaching beyond standard BOW contexts towards contexts based on dependency parses (Bansal et al., 2014; Melamud et al., 2016) or symmetric patterns (Schwartz et al., 2015, 2016) yields significant improvements in learning representations for particular word classes such as *adjectives (A)* and *verbs (V)*. Moreover, Schwartz et al. (2016) demonstrated that a subset of dependency-based contexts which covers only coordination structures is particularly effective for SGNS training, both in terms of the quality of the induced representations and in the reduced training time of the model. Interestingly, they also demonstrated that despite the success with adjectives and verbs, BOW contexts are still the optimal choice when learning representations for *nouns (N)*.

In this work, we propose a simple yet effective framework for selecting *context configurations*, which yields improved representations for verbs, adjectives, *and* nouns. We start with a definition of our context configuration space (Sect. 3.1). Our basic definition of a context refers to a single typed (or labeled) dependency link between words (e.g., the `amod` link or the `dobj` link). Our configuration space then naturally consists of all possible subsets of the set of labeled dependency links between words. We employ the universal dependencies (UD) scheme to make our framework applicable across

languages. We then describe (Sect. 3.2) our adapted beam search algorithm that aims to select an optimal context configuration for a given word class.

We show that SGNS requires different context configurations to produce improved results for each word class. For instance, our algorithm detects that the combination of `amod` and `conj` contexts is effective for adjective representation. Moreover, some contexts that boost representation learning for one word class (e.g., `amod` contexts for adjectives) may be uninformative when learning representations for another class (e.g., `amod` for verbs). By removing such dispensable contexts, we are able both to speed up the SGNS training and to improve representation quality.

We first experiment with the task of predicting similarity scores for the A/V/N portions of the benchmarking SimLex-999 evaluation set, running our algorithm in a standard SGNS experimental setup (Levy et al., 2015). When training SGNS with our learned context configurations it outperforms SGNS trained with the best previously proposed context type *for each word class*: the improvements in Spearman's ρ rank correlations are 6 (A), 6 (V), and 5 (N) points. We also show that by building context configurations we obtain improvements on the entire SimLex-999 (4 ρ points over the best baseline). Interestingly, this context configuration is not the optimal configuration for any word class.

We then demonstrate that our approach is robust by showing that transferring the optimal configurations learned in the above setup to three other setups yields improved performance. First, the above context configurations, learned with the SGNS training on the English Wikipedia corpus, have an even stronger impact on SimLex999 performance when SGNS is trained on a larger corpus. Second, the transferred configurations also result in competitive performance on the task of solving class-specific TOEFL questions. Finally, we transfer the learned context configurations across languages: these configurations improve the SGNS performance when trained with German or Italian corpora and evaluated on class-specific subsets of the multilingual SimLex-999 (Leviant and Reichart, 2015), without any language-specific tuning.

2 Related Work

Word representation models typically train on (*word, context*) pairs. Traditionally, most models use bag-of-words (BOW) contexts, which represent a word using its neighbouring words, irrespective of the syntactic or semantic relations between them (Collobert et al., 2011; Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014, inter alia). Several alternative context types have been proposed, motivated by the limitations of BOW contexts, most notably their focus on topical rather than functional similarity (e.g., *coffee:cup* vs. *coffee:tea*). These include dependency contexts (Padó and Lapata, 2007; Levy and Goldberg, 2014a), pattern contexts (Baroni et al., 2010; Schwartz et al., 2015) and substitute vectors (Yatbaz et al., 2012; Melamud et al., 2015).

Several recent studies examined the effect of context types on word representation learning. Melamud et al. (2016) compared three context types on a set of intrinsic and extrinsic evaluation setups: BOW, dependency links, and substitute vectors. They show that the optimal type largely depends on the task at hand, with dependency-based contexts displaying strong performance on semantic similarity tasks. Vulić and Korhonen (2016) extended the comparison to more languages, reaching similar conclusions. Schwartz et al. (2016), showed that symmetric patterns are useful as contexts for V and A similarity, while BOW still works best for nouns. They also indicated that coordination structures, a particular dependency link, are more useful for verbs and adjectives than the entire set of dependencies. In this work, we generalise their approach: our algorithm systematically and efficiently searches the space of dependency-based context configurations, yielding *class-specific* representations with substantial gains *for all three word classes*.

Previous attempts on specialising word representations for a particular relation (e.g., similarity vs relatedness, antonyms) operate in one of two frameworks: (1) modifying the prior or the regularisation of the original training procedure (Yu and Dredze, 2014; Wieting et al., 2015; Liu et al., 2015; Kiela et al., 2015; Ling et al., 2015b); (2) post-processing procedures which use lexical knowledge to refine previously trained word vectors (Faruqui et al., 2015; Wieting et al., 2015; Mrkšić et al., 2017). Our work suggests that the induced representations can be specialised by directly training the word representation model with carefully selected contexts.

3 Context Selection: Methodology

The goal of our work is to develop a methodology for the identification of optimal context configura-

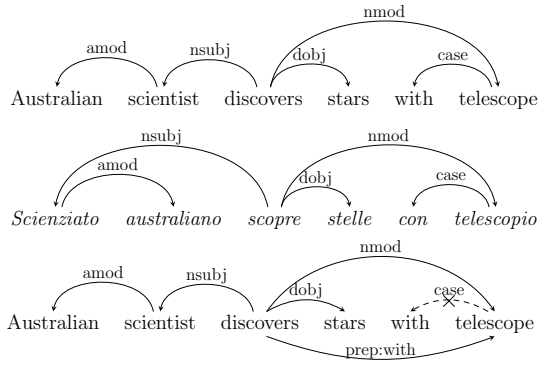

Figure 1: Extracting dependency-based contexts. **Top**: An example English sentence from (Levy and Goldberg, 2014a), now UD-parsed. **Middle**: the same sentence in Italian, UD-parsed. Note the similarity between the two parses which suggests that our context selection framework may be extended to other languages. **Bottom**: prepositional arc collapsing. The uninformative short-range `case` arc is removed, while a "pseudo-arc" specifying the exact link (`prep:with`) between *discovers* and *telescope* is added.

tions for word representation model training. We hope to get improved word representations and, at the same time, cut down the training time of the word representation model. Fundamentally, we are not trying to design a new word representation model, but rather to find valuable configurations for existing algorithms.

The motivation to search for such training context configurations lies in the intuition that the distributional hypothesis (Harris, 1954) should not necessarily be made with respect to BOW contexts. Instead, it may be restated as a series of statements according to particular word relations. For example, the hypothesis can be restated as: "two adjectives are similar if they modify similar nouns", which is captured by the `amod` typed dependency relation. This could also be reversed to reflect noun similarity by saying that "two nouns are similar if they are modified by similar adjectives". In another example, "two verbs are similar if they are used as predicates of similar nominal subjects" (the `nsubj` and `nsubjpass` dependency relations).

First, we have to define an expressive context configuration space that contains potential training configurations and is effectively decomposed so that useful configurations may be sought algorithmically. We can then continue by designing a search algorithm over the configuration space.

3.1 Context Configuration Space

We focus on the configuration space based on dependency-based contexts (DEPS) (Padó and Lapata, 2007; Utt and Padó, 2014). We choose this space due to multiple reasons. First, dependency structures are known to be very useful in capturing functional relations between words, even if these relations are long distance. Second, they have been proven useful in learning word embeddings (Levy and Goldberg, 2014a; Melamud et al., 2016). Finally, owing to the recent development of the Universal Dependencies (UD) annotation scheme (McDonald et al., 2013; Nivre et al., 2016)[1] it is possible to reason over dependency structures in a multilingual manner (e.g., Fig. 1). Consequently, a search algorithm in such DEPS-based configuration space can be developed for multiple languages based on the same design principles. Indeed, in this work we show that the optimal configurations for English translate to improved representations in two additional languages, German and Italian.

And so, given a (UD-)parsed training corpus, for each target word w with modifiers $m_1, \ldots, m_k$ and a head h, the word w is paired with context elements $m_1\_r_1, \ldots, m_k\_r_k, h\_r_h^{-1}$, where r is the type of the dependency relation between the head and the modifier (e.g., `amod`), and r^{-1} denotes an inverse relation. To simplify the presentation, we adopt the assumption that all training data for the word representation model are in the form of such $(word, context)$ pairs (Levy and Goldberg, 2014a,c), where $word$ is the current target word, and $context$ is its observed context (e.g., BOW, positional, dependency-based). A naive version of DEPS extracts contexts from the parsed corpus without any post-processing. Given the example from Fig. 1, the DEPS contexts of *discovers* are: *scientist_nsubj*, *stars_dobj*, *telescope_nmod*.

DEPS not only emphasises functional similarity, but also provides a natural implicit grouping of related contexts. For instance, all pairs with the shared relation r and r^{-1} are taken as an r-based *context bag*, e.g., the pairs {*(scientist, Australian_amod)*, *(Australian, scientist_amod^{-1})*} from Fig. 1 are inserted into the `amod` context bag, while {*(discovers, stars_dobj)*, *(stars, discovers_dobj^{-1})*} are labelled with `dobj`.

Assume that we have obtained M distinct dependency relations $r_1, \ldots, r_M$ after parsing and post-processing the corpus. The j-th *individual context*

[1] http://universaldependencies.org/ (V1.4 used)

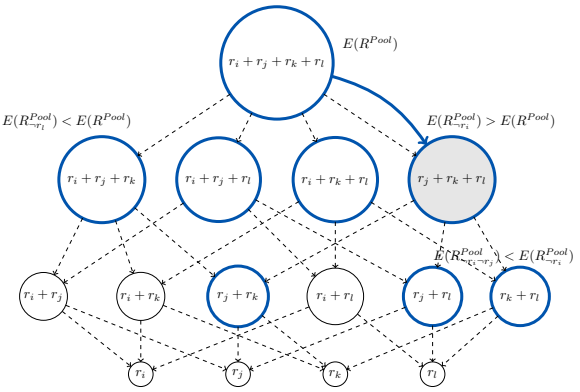

Figure 2: An illustration of Alg. 1. The search space is presented as a DAG with direct links between origin configurations (e.g., $r_i + r_j + r_k$) and all its children configurations obtained by removing exactly one individual bag from the origin (e.g., $r_i + r_j$, $r_j + r_k$). After automatically constructing the initial pool (line 1), the entry point of the algorithm is the R^{Pool} configuration (line 2). Thicker blue circles denote visited configurations, while the gray circle denotes the best configuration found.

bag, $j = 1, \ldots, M$, labelled r_j, is a bag (or a multiset) of $(word, context)$ pairs where $context$ has one of the following forms: v_r_j or $v\_r_j^{-1}$, where v is some vocabulary word. A *context configuration* is then simply a set of individual context bags, e.g., $R = \{r_i, r_j, r_k\}$, also labelled as R: $r_i + r_j + r_k$. We call a configuration consisting of K individual context bags a K-set configuration (e.g., in this example, R is a 3-set configuration).[2]

Although a brute-force exhaustive search over all possible configurations is possible in theory and for small pools (e.g., for adjectives, see Tab. 2), it becomes challenging or practically infeasible for large pools and large training data. For instance, based on the pool from Tab. 2, the search for the optimal configuration would involve trying out $2^{10} - 1 = 1023$ configurations for nouns (i.e., training 1023 different word representation models). Therefore, to reduce the number of visited configurations, we present a simple heuristic search algorithm inspired by beam search (Pearl, 1984).

[2]A note on the nomenclature and notation: Each context configuration may be seen as a set of context bags, as it does not allow for repetition of its constituent context bags. For simplicity and clarity of presentation, we use dependency relation types (e.g., r_i = `amod`, r_j = `acl`) as labels for context bags. The reader has to be aware that a configuration $R = \{r_i, r_j, r_k\}$ is not by any means a set of relation types/names, but is in fact a multiset of all $(word, context)$ pairs belonging to the corresponding context bags labelled with r_i, r_j, r_k.

Algorithm 1: Best Configuration Search

Input : Set of M individual context bags: $S = \{r'_1, r'_2, \ldots, r'_M\}$

1 **build**: *pool* of those $K \leq M$ candidate individual context bags $\{r_1, \ldots, r_K\}$ for which $E(r_i) >= threshold, i \in \{1, \ldots, M\}$, where $E(\cdot)$ is a fitness function.
2 **build:** K-set configuration $R^{Pool} = \{r_1, \ldots, r_K\}$;
3 **initialize:** (1) set of candidate configurations $\mathbf{R} = \{R^{Pool}\}$; (2) current level $l = K$; (3) best configuration $R_o = \emptyset$;
4 **search**:
5 **repeat**
6 $\mathbf{R}_n \leftarrow \emptyset$;
7 $R_o \leftarrow \underset{R \in \mathbf{R} \cup \{R_o\}}{\arg\max} E(R)$;
8 **foreach** $R \in \mathbf{R}$ **do**
9 **foreach** $r_i \in R$ **do**
10 **build** new $(l-1)$-set context configuration $R_{\neg r_i} = R - \{r_i\}$;
11 **if** $E(R_{\neg r_i}) \geq E(R)$ **then**
12 $\mathbf{R_n} \leftarrow \mathbf{R_n} \cup \{R_{\neg r_i}\}$;
13 $l \leftarrow l - 1$;
14 $\mathbf{R} \leftarrow \mathbf{R_n}$;
15 **until** $l == 0$ *or* $\mathbf{R} == \emptyset$;

Output : Best configuration R_o

3.2 Class-Specific Configuration Search

Alg. 1 provides a high-level overview of the algorithm. An example of its flow is given in Fig. 2. Starting from S, the set of all possible M individual context bags, the algorithm automatically detects the subset $S_K \subseteq S$, $|S_K| = K$, of candidate individual bags that are used as the initial pool (line 1 of Alg. 1). The selection is based on some fitness (goal) function E. In our setup, $E(R)$ is Spearman's ρ correlation with human judgment scores obtained on the development set after training the word representation model with the configuration R. The selection step relies on a simple threshold: we use a threshold of $\rho \geq 0.2$ without any fine-tuning in all experiments with all word classes.

We find this step to facilitate efficiency at a minor cost for accuracy. For example, since `amod` denotes an adjectival modifier of a noun, an efficient search procedure may safely remove this bag from the pool of candidate bags for verbs.

The search algorithm then starts from the full K-set R^{Pool} configuration (line 3) and tests K $(K-1)$-set configurations where exactly one individual bag r_i is removed to generate each such configuration (line 10). It then retains only the set of configurations that score higher than the origin K-set configuration (lines 11-12, see Fig. 2). Using this principle, it continues searching only over lower-level $(l-1)$-set configurations that further

improve performance over their l-set origin configuration. It stops if it reaches the lowest level or if it cannot improve the goal function any more (line 15). The best scoring configuration is returned (n.b., not guaranteed to be the global optimum).

In our experiments with this heuristic, the search for the optimal configuration for verbs is performed only over 13 1-set configurations plus 26 other configurations (39 out of 133 possible configurations).[3] For nouns, the advantage of the heuristic is even more dramatic: only 104 out of 1026 possible configurations were considered during the search.[4]

4 Experimental Setup

4.1 Implementation Details

Word Representation Model We experiment with SGNS (Mikolov et al., 2013), the standard and very robust choice in vector space modeling (Levy et al., 2015). In all experiments we use `word2vecf`, a reimplementation of `word2vec` able to learn from arbitrary $(word, context)$ pairs.[5] For details concerning the implementation, we refer the reader to (Goldberg and Levy, 2014; Levy and Goldberg, 2014a).

The SGNS preprocessing scheme was replicated from (Levy and Goldberg, 2014a; Levy et al., 2015). After lowercasing, all words and contexts that appeared less than 100 times were filtered. When considering all dependency types, the vocabulary spans approximately 185K word types.[6] Further, all representations were trained with $d = 300$ (very similar trends are observed with $d = 100, 500$).

The same setup was used in prior work (Schwartz et al., 2016; Vulić and Korhonen, 2016). Keeping the representation model fixed across experiments and varying only the context type allows us to attribute any differences in results to a sole factor: the context type. We plan to experiment with other representation models in future work.

[3]The total is 133 as we have to include 6 additional 1-set configurations that have to be tested (line 1 of Alg. 1) but are not included in the initial pool for verbs (line 2).

[4]We also experimented with a less conservative variant which does not stop when lower-level configurations do not improve E; it instead follows the path of the best-scoring lower-level configuration even if its score is lower than that of its origin. As we do not observe any significant improvement with this variant, we opt for the faster and simpler one.

[5]https://bitbucket.org/yoavgo/word2vecf

[6]SGNS for all models was trained using stochastic gradient descent and standard settings: 15 negative samples, global learning rate: 0.025, subsampling rate: $1e-4$, 15 epochs.

Universal Dependencies as Labels The adopted UD scheme leans on the universal Stanford dependencies (de Marneffe et al., 2014) complemented with the universal POS tagset (Petrov et al., 2012). It is straightforward to "translate" previous annotation schemes to UD (de Marneffe et al., 2014). Providing a consistently annotated inventory of categories for similar syntactic constructions across languages, the UD scheme facilitates representation learning in languages other than English, as shown in (Vulić and Korhonen, 2016; Vulić, 2017).

Individual Context Bags Standard post-parsing steps are performed in order to obtain an initial list of individual context bags for our algorithm: (1) Prepositional arcs are collapsed ((Levy and Goldberg, 2014a; Vulić and Korhonen, 2016), see Fig. 1). Following this procedure, all pairs where the relation r has the form `prep:X` (where `X` is a preposition) are subsumed to a context bag labelled `prep`; (2) Similar labels are merged into a single label (e.g., direct (`dobj`) and indirect objects (`iobj`) are merged into `obj`); (3) Pairs with infrequent and uninformative labels are removed (e.g., `punct`, `goeswith`, `cc`).

Coordination-based contexts are extracted as in prior work (Schwartz et al., 2016), distinguishing between left and right contexts extracted from the `conj` relation; the label for this bag is `conjlr`. We also utilise the variant that does not make the distinction, labeled `conjll`. If both are used, the label is simply `conj=conjlr+conjll`.[7]

Consequently, the individual context bags we use in all experiments are: `subj`, `obj`, `comp`, `nummod`, `appos`, `nmod`, `acl`, `amod`, `prep`, `adv`, `compound`, `conjlr`, `conjll`.

4.2 Training and Evaluation

We run the algorithm for context configuration selection only once, with the SGNS training setup described below. Our main evaluation setup is presented below, but the learned configurations are tested in additional setups, detailed in Sect. 5.

Training Data Our training corpus is the cleaned and tokenised English Polyglot Wikipedia data (Al-Rfou et al., 2013),[8] consisting of approxi-

[7]Given the coordination structure *boys and girls*, `conjlr` training pairs are *(boys, girls_conj), (girls, boys_conj^{-1})*, while `conjll` pairs are *(boys, girls_conj), (girls, boys_conj)*.

[8]https://sites.google.com/site/rmyeid/projects/polyglot

mately 75M sentences and 1.7B word tokens. The Wikipedia data were POS-tagged with universal POS (UPOS) tags (Petrov et al., 2012) using the state-of-the art TurboTagger (Martins et al., 2013).[9] The parser was trained using default settings (SVM MIRA with 20 iterations, no further parameter tuning) on the TRAIN+DEV portion of the UD treebank annotated with UPOS tags. The data were then parsed with UD using the graph-based Mate parser v3.61 (Bohnet, 2010)[10] with standard settings on TRAIN+DEV of the UD treebank.

Evaluation We experiment with the verb pair (222 pairs), adjective pair (111 pairs), and noun pair (666 pairs) portions of SimLex-999. We report Spearman's ρ correlation between the ranks derived from the scores of the evaluated models and the human scores. Our evaluation setup is borrowed from Levy et al. (2015): we perform 2-fold cross-validation, where the context configurations are optimised on a development set, separate from the unseen test data. Unless stated otherwise, the reported scores are always the averages of the 2 runs, computed in the standard fashion by applying the cosine similarity to the vectors of words participating in a pair.

4.3 Baselines

Baseline Context Types We compare the context configurations found by Alg. 1 against baseline contexts from prior work:

- **BOW**: Standard bag-of-words contexts.
- **POSIT**: Positional contexts (Schütze, 1993; Levy and Goldberg, 2014b; Ling et al., 2015a), which enrich BOW with information on the sequential position of each context word. Given the example from Fig. 1, POSIT with the window size 2 extracts the following contexts for *discovers*: *Australian_-2*, *scientist_-1*, *stars_+2*, *with_+1*.
- **DEPS-All**: All dependency links without any context selection, extracted from dependency-parsed data with prepositional arc collapsing.
- **COORD**: Coordination-based contexts are used as fast lightweight contexts for improved representations of adjectives and verbs (Schwartz et al., 2016). This is in fact the `conjlr` context bag, a subset of DEPS-All.
- **SP**: Contexts based on symmetric patterns (SPs, (Davidov and Rappoport, 2006; Schwartz et al., 2015)). For example, if the word X and the word Y appear in the lexico-syntactic symmetric pattern "X or Y" in the SGNS training corpus, then Y is an SP context instance for X, and vice versa.

The development set was used to tune the window size for BOW and POSIT (to 2) and the parameters of the SP extraction algorithm.[11]

Baseline Greedy Search Algorithm We also compare our search algorithm to its greedy variant: at each iteration of lines 8-12 in Alg. 1, R_n now keeps only the best configuration of size $l-1$ that perform better than the initial configuration of size l, instead of all such configurations.

Context Group	Adj	Verb	Noun
`conjlr` (A+N+V)	0.415	0.281	0.401
`obj` (N+V)	-0.028	0.309	0.390
`prep` (N+V)	0.188	0.344	0.387
`amod` (A+N)	0.479	0.058	0.398
`compound` (N)	-0.124	-0.019	0.416
`adv` (V)	0.197	0.342	0.104
`nummod` (-)	-0.142	-0.065	0.029

Table 1: 2-fold cross-validation results for an illustrative selection of individual context bags. Results are presented for the noun, verb and adjective subsets of SimLex-999. Values in parentheses denote the class-specific initial pools to which each context is selected based on its ρ score (line 1 of Alg. 1).

Adjectives	Verbs	Nouns
`amod, conjlr, conjll`	`prep, acl, obj, comp, adv, conjlr, conjll`	`amod, prep, compound, subj, obj, appos, acl, nmod, conjlr, conjll`

Table 2: Automatically constructed initial pools of candidate bags for each word class (Sect. 3.2).

5 Results and Discussion

5.1 Main Evaluation Setup

Not All Context Bags are Created Equal First, we test the performance of *individual* context bags across SimLex-999 adjective, verb, and noun subsets. Besides providing insight on the intuition behind context selection, these findings are important for the automatic selection of class-specific pools (line 1 of Alg. 1). The results are shown in Tab. 1.

The experiment supports our intuition (see Sect. 3.2): some context bags are definitely not useful for some classes and may be safely removed

[9] http://www.cs.cmu.edu/~ark/TurboParser/

[10] https://code.google.com/archive/p/mate-tools/

[11] The SP extraction algorithm is available online: homes.cs.washington.edu/~roysch/software/dr06/dr06.html

Baselines	(Verbs)
BOW (win=2)	0.336
POSIT (win=2)	0.345
COORD (conjlr)	0.283
SP	0.349
DEPS-All	0.344
Configurations: Verbs	
POOL-ALL	0.379
prep+acl+obj+adv+conj	0.393
prep+acl+obj+comp+conj	0.344
prep+obj+comp+adv+conj	*0.391*†
prep+acl+adv+conj (BEST)	**0.409**
prep+acl+obj+adv	0.392
prep+acl+adv	0.407
prep+acl+conj	0.390
acl+obj+adv+conj	0.345
acl+obj+adv	0.385

Baselines	(Nouns)
BOW (win=2)	0.435
POSIT (win=2)	0.437
COORD (conjlr)	0.392
SP	0.372
DEPS-All	0.441
Configurations: Nouns	
POOL-ALL	0.469
amod+subj+obj+appos+compound+nmod+conj	0.478
amod+subj+obj+appos+compound+conj	0.487
amod+subj+obj+appos+compound+conjlr	*0.476*†
amod+subj+obj+compound+conj (BEST)	**0.491**
amod+subj+obj+appos+conj	0.470
subj+obj+compound+conj	0.479
amod+subj+compound+conj	0.481
amod+subj+obj+compound	0.478
amod+obj+compound+conj	0.481

Table 3: Results on the SimLex-999 test data over (a) **verbs** and (b) **nouns** subsets. Only a selection of context configurations optimised for verb and noun similarity are shown. POOL-ALL denotes a configuration where all individual context bags from the verbs/nouns-oriented pools (see Table 2) are used. BEST denotes the best performing configuration found by Alg. 1. Other configurations visited by Alg. 1 that score higher than the best scoring baseline context type for each word class are in gray. Scores obtained using a greedy search algorithm instead of Alg. 1 are in italic, marked with a cross (†).

Baselines	(Adjectives)
BOW (win=2)	0.489
POSIT (win=2)	0.460
COORD (conjlr)	0.407
SP	0.395
DEPS-All	0.360
Configurations: Adjectives	
POOL-ALL: amod+conj (BEST)	**0.546**†
amod+conjlr	0.527
amod+conjll	0.531
conj	0.470

Table 4: Results on the SimLex-999 **adjectives** subset with adjective-specific configurations.

when performing the class-specific SGNS training. For instance, the amod bag is indeed important for adjective and noun similarity, and at the same time it does not encode any useful information regarding verb similarity. compound is, as expected, useful only for nouns. Tab. 1 also suggests that some context bags (e.g., nummod) do not encode any informative contextual evidence regarding similarity, therefore they can be discarded. The initial results with individual context bags help to reduce the pool of candidate bags (line 1 in Alg. 1), see Tab. 2.

Searching for Improved Configurations Next, we test if we can improve class-specific representations by selecting class-specific configurations. Results are summarised in Tables 3 and 4. Indeed, class-specific configurations yield better representations, as is evident from the scores: the improvements with the best class-specific configurations found by Alg. 1 are approximately 6 ρ points for adjectives, 6 points for verbs, and 5 points for nouns over the best baseline for each class.

The improvements are visible even with configurations that simply pool all candidate individual bags (POOL-ALL), without running Alg. 1 beyond line 1. However, further careful context selection, i.e., traversing the configuration space using Alg. 1 leads to additional improvements for V and N (gains of 3 and 2.2 ρ points). Very similar improved scores are achieved with a variety of configurations (see Tab. 3), especially in the neighbourhood of the best configuration found by Alg. 1. This indicates that the method is quite robust: even sub-optimal[12] solutions result in improved class-specific representations. Furthermore, our algorithm is able to find better configurations for verbs and nouns compared to its greedy variant. Finally, our algorithm generalises well: the best scoring configuration on the dev set is always the best one on the test set.

Training: Fast and/or Accurate? Carefully selected configurations are also likely to reduce SGNS training times. Indeed, the configuration-based model trains on only 14% (A), 26.2% (V), and 33.6% (N) of all dependency-based contexts. The training times and statistics for each context type are displayed in Tab. 5. All models

[12]The term *optimal* here and later in the text refers to the best configuration returned by our algorithm.

Context Type	Training Time	# Pairs
BOW (win=2)	179mins 27s	5.974G
POSIT (win=2)	190mins 12s	5.974G
COORD (conjlr)	4mins 11s	129.69M
SP	1mins 29s	46.37M
DEPS-All	103mins 35s	3.165G
BEST-ADJ	14mins 5s	447.4M
BEST-VERBS	29mins 48s	828.55M
BEST-NOUNS	41mins 14s	1.063G

Table 5: Training time (wall-clock time reported) in minutes for SGNS ($d = 300$) with different context types. BEST-* denotes the best scoring configuration for each class found by Alg. 1. *#Pairs* shows a total number of pairs used in SGNS training for each context type.

Context Type	Adj	Verbs	Nouns	All
BOW (win=2)	0.604	0.307	0.501	0.464
POSIT (win=2)	0.585	0.400	0.471	0.469
COORD (conjlr)	0.629	0.413	0.428	0.430
SP	0.649	**0.458**	0.414	0.444
DEPS-All	0.574	0.389	0.492	0.464
BEST-ADJ	**0.671**	0.348	0.504	0.449
BEST-VERBS	0.392	0.455	0.478	0.448
BEST-NOUNS	0.581	0.327	**0.535**	0.489
BEST-ALL	0.616	0.402	0.519	**0.506**

Table 6: Results on the A/V/N SimLex-999 subsets, and on the entire set (*All*) in the setup from Schwartz et al. (2016). $d = 500$. BEST-* are again the best class-specific configs returned by Alg. 1.

were trained using parallel training on 10 Intel(R) Xeon(R) E5-2667 2.90GHz processors. The results indicate that class-specific configurations are not as lightweight and fast as SP or COORD contexts (Schwartz et al., 2016). However, they also suggest that such configurations provide a good balance between accuracy and speed: they reach peak performances for each class, outscoring all baseline context types (including SP and COORD), while training is still much faster than with "heavyweight" context types such as BOW, POSIT or DEPS-All.

Now that we verified the decrease in training time our algorithm provides for the final training, it makes sense to ask whether the configurations it finds are valuable *in other setups*. This will make the fast training of practical importance.

5.2 Generalisation: Configuration Transfer

Another Training Setup We first test whether the context configurations learned in Sect. 5.1 are useful when SGNS is trained in another English setup (Schwartz et al., 2016), with more training data and other annotation and parser choices, while evaluation is still performed on SimLex-999.

In this setup the training corpus is the 8B words corpus generated by the word2vec script.[13] A preprocessing step now merges common word pairs and triplets to expression tokens (e.g., *Bilbo_Baggins*). The corpus is parsed with labelled Stanford dependencies (de Marneffe and Manning, 2008) using the Stanford POS Tagger (Toutanova et al., 2003) and the stack version of the MALT parser (Goldberg and Nivre, 2012). SGNS preprocessing and parameters are also replicated; we now train 500-dim embeddings as in prior work.[14]

Results are presented in Tab. 6. The imported class-specific configurations, computed using a much smaller corpus (Sect. 5.1), again outperform competitive baseline context types for adjectives and nouns. The BEST-VERBS configuration is outscored by SP, but the margin is negligible. We also evaluate another configuration found using Alg. 1 in Sect. 5.1, which targets the overall improved performance without any finer-grained division to classes (BEST-ALL). This configuration (amod+subj+obj+compound+prep+adv+conj) outperforms all baseline models on the entire benchmark. Interestingly, the non-specific BEST-ALL configuration falls short of A/V/N-specific configurations for each class. This unambiguously implies that the "trade-off" configuration targeting all three classes at the same time differs from specialised class-specific configurations.

Experiments on Other Languages We next test whether the optimal context configurations computed in Sect. 5.1 with English training data are also useful for other languages. For this, we train SGNS models on the Italian (IT) and German (DE) Polyglot Wikipedia corpora with those configurations, and evaluate on the IT and DE multilingual SimLex-999 (Leviant and Reichart, 2015).[15]

Our results demonstrate similar patterns as for English, and indicate that our framework can be easily applied to other languages. For instance, the BEST-ADJ configuration (the same configuration as in Tab. 4 and Tab. 7) yields an improvement of 8

[13] code.google.com/p/word2vec/source/browse/trunk/

[14] The "translation" from labelled Stanford dependencies into UD is performed using the mapping from de Marneffe et al. (2014), e.g., nn is mapped into compound, and rcmod, partmod, infmod are all mapped into one bag: acl.

[15] http://leviants.com/ira.leviant/MultilingualVSMdata.html

Context Type	Adj-Q	Verb-Q	Noun-Q
BOW (win=2)	31/41	14/19	16/19
POSIT (win=2)	**32/41**	13/19	15/19
COORD (conjlr)	26/41	11/19	8/19
SP	26/41	11/19	12/19
DEPS-All	31/41	14/19	16/19
BEST-ADJ	**32/41**	12/19	15/19
BEST-VERBS	24/41	**15/19**	16/19
BEST-NOUNS	30/41	14/19	**17/19**

Table 7: Results on the A/V/N TOEFL question subsets. The reported scores are in the following form: *correct_answers/overall_questions*. *Adj-Q* refers to the subset of TOEFL questions targeting adjectives; similar for *Verb-Q* and *Noun-Q*. BEST-* refer to the best class-specific configurations from Tab. 3 and Tab. 4.

ρ points and 4 ρ points over the strongest adjectives baseline in IT and DE, respectively. We get similar improvements for nouns (IT: 3 ρ points, DE: 2 ρ points), and verbs (IT: 2, DE: 4).

TOEFL Evaluation We also verify that the selection of class-specific configurations (Sect. 5.1) is useful beyond the core SimLex evaluation. For this aim, we evaluate on the A, V, and N TOEFL questions (Landauer and Dumais, 1997). The results are summarised in Tab. 7. Despite the limited size of the TOEFL dataset, we observe positive trends in the reported results (e.g., V-specific configurations yield a small gain on verb questions), showcasing the potential of class-specific training in this task.

6 Conclusion and Future Work

We have presented a novel framework for selecting class-specific context configurations which yield improved representations for prominent word classes: adjectives, verbs, and nouns. Its design and dependence on the Universal Dependencies annotation scheme makes it applicable in different languages. We have proposed an algorithm that is able to find a suitable class-specific configuration while making the search over the large space of possible context configurations computationally feasible. Each word class requires a different class-specific configuration to produce improved results on the class-specific subset of SimLex-999 in English, Italian, and German. We also show that the selection of context configurations is robust as once learned configuration may be effectively transferred to other data setups, tasks, and languages without additional retraining or fine-tuning.

In future work, we plan to test the framework with finer-grained contexts, investigating beyond POS-based word classes and dependency links. Exploring more sophisticated algorithms that can efficiently search richer configuration spaces is also an intriguing direction. Another research avenue is application of the context selection idea to other representation models beyond SGNS tested in this work, and experimenting with assigning weights to context subsets. Finally, we plan to test the portability of our approach to more languages.

Acknowledgments

This work is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909). Roy Schwartz was supported by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI). The authors are grateful to the anonymous reviewers for their helpful and constructive suggestions.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *CoNLL*. pages 183–192. http://www.aclweb.org/anthology/W13-3520.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *ACL*. pages 809–815. http://www.aclweb.org/anthology/P14-2131.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL*. pages 238–247. http://www.aclweb.org/anthology/P14-1023.

Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A corpus-based semantic model based on properties and types. *Cognitive Science* pages 222–254. https://doi.org/10.1111/j.1551-6709.2009.01068.x.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*. pages 89–97. http://www.aclweb.org/anthology/C10-1011.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*. pages 740–750. http://www.aclweb.org/anthology/D14-1082.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and

Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537. http://dl.acm.org/citation.cfm?id=1953048.2078186.

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *ACL*. pages 297–304. http://www.aclweb.org/anthology/P06-1038.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *LREC*. pages 4585–4592. http://www.lrec-conf.org/proceedings/lrec2014/summaries/1062.html.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*. pages 1–8. http://www.aclweb.org/anthology/W08-1301.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *NAACL-HLT*. pages 1606–1615. http://www.aclweb.org/anthology/N15-1184.

Yoav Goldberg and Omer Levy. 2014. Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR* abs/1402.3722. http://arxiv.org/abs/1402.3722.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *COLING*. pages 959–976. http://www.aclweb.org/anthology/C12-1059.

Zellig S. Harris. 1954. Distributional structure. *Word* 10(23):146–162. https://doi.org/10.1080/00437956.1954.11659520.

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *EMNLP*. pages 2044–2048. http://aclweb.org/anthology/D15-1242.

Thomas K. Landauer and Susan T. Dumais. 1997. Solutions to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104(2):211–240. https://doi.org/10.1037/0033-295X.104.2.211.

Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *CoRR* abs/1508.00106. http://arxiv.org/abs/1508.00106.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *ACL*. pages 302–308. http://www.aclweb.org/anthology/P14-2050.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *CoNLL*. pages 171–180. http://www.aclweb.org/anthology/W14-1618.

Omer Levy and Yoav Goldberg. 2014c. Neural word embedding as implicit matrix factorization. In *NIPS*. pages 2177–2185. http://papers.nips.cc/paper/5477.pdf.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the ACL* 3:211–225.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015a. Two/too simple adaptations of Word2Vec for syntax problems. In *NAACL-HLT*. pages 1299–1304. http://www.aclweb.org/anthology/N15-1142.

Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. 2015b. Not all contexts are created equal: Better word representations with variable attention. In *EMNLP*. pages 1367–1372. http://aclweb.org/anthology/D15-1161.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. In *ACL*. pages 1501–1511. http://www.aclweb.org/anthology/P15-1145.

André F. T. Martins, Miguel B. Almeida, and Noah A. Smith. 2013. Turning on the Turbo: Fast third-order non-projective turbo parsers. In *ACL*. pages 617–622. http://www.aclweb.org/anthology/P13-2109.

Ryan T. McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B. Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *ACL*. pages 92–97. http://www.aclweb.org/anthology/P13-2017.

Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015. Modeling word meaning in context with substitute vectors. In *NAACL-HLT*. pages 472–482. http://www.aclweb.org/anthology/N15-1050.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *NAACL-HLT*. http://www.aclweb.org/anthology/N16-1118.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. pages 3111–3119.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*. pages 2265–2273.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialisation of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the ACL* https://arxiv.org/abs/1706.00374.

Joakim Nivre et al. 2016. Universal Dependencies 1.4. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* 33(2):161–199. https://doi.org/10.1162/coli.2007.33.2.161.

Judea Pearl. 1984. Heuristics: Intelligent search strategies for computer problem solving .

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2012. A universal part-of-speech tagset. In *LREC*. pages 2089–2096. http://www.lrec-conf.org/proceedings/lrec2012/summaries/274.html.

Hinrich Schütze. 1993. Part-of-speech induction from scratch. In *ACL*. pages 251–258. http://www.aclweb.org/anthology/P93-1034.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *CoNLL*. pages 258–267. http://www.aclweb.org/anthology/K15-1026.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2016. Symmetric patterns and coordinations: Fast and enhanced representations of verbs and adjectives. In *NAACL-HLT*. pages 499–505. http://www.aclweb.org/anthology/N16-1060.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL-HLT*. pages 173–180. http://aclweb.org/anthology/N/N03/.

Joseph P. Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*. pages 384–394. http://www.aclweb.org/anthology/P10-1040.

Jason Utt and Sebastian Padó. 2014. Crosslingual and multilingual construction of syntax-based vector space models. *Transactions of the ACL* 2:245–258.

Ivan Vulić. 2017. Cross-lingual syntactically informed distributed word representations. In *EACL*. pages 408–414. http://www.aclweb.org/anthology/E17-2065.

Ivan Vulić and Anna Korhonen. 2016. Is "universal syntax" universally useful for learning distributed word representations? In *ACL*. pages 518–524. http://anthology.aclweb.org/P16-2084.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the ACL* 3:345–358. http://aclweb.org/anthology/Q15-1025.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *EMNLP*. pages 940–951. http://www.aclweb.org/anthology/D12-1086.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *ACL*. pages 545–550. http://www.aclweb.org/anthology/P14-2089.